Lesson 1

Introduction to Microprocessor Systems

INTRODUCTION

The microprocessor is a large, complex integrated circuit (IC) containing all the computation and control circuitry for a small computer. It provides economical computing power for many devices, including "smart" voltmeters, microwave ovens, cash registers, and even games. In this lesson, the general operation of microprocessor systems and the building blocks which are used are described.

THE DEVELOPMENT OF THE MICROPROCESSOR

The earliest electronic computers were built using thousands of vacuum tubes. These machines were extremely large and unreliable, and were mostly a laboratory curiosity. The next generation was built with transistors, which made computers much more reliable and reduced their size and cost. These "solid-state" machines marked the beginning of the computer as a practical device.

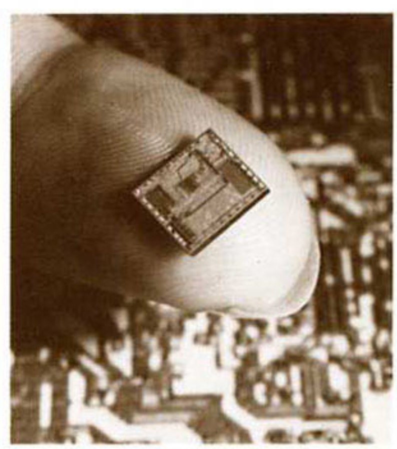
ENIAC Calculating Machine. *Built in 1946, it was the first general-purpose electronic computing machine. It used over 18,000 vacuum tubes and required a power supply half the size of the computer itself. (Photo Courtesy UPI)*

In the 1960s smaller, more powerful computers were built using hundreds of gates, flip-flops, and other similar integrated circuits. These ICs are called *Small Scale Integration* (SSI) devices. As semiconductor technology developed, it became possible to put dozens of gates on a single IC. Examples of these *Medium Scale Integration* (MSI) ICs are counters, decoders, registers, and adders.

Programmable Desktop Calculator. *Introduced in 1968, it is more powerful than the much larger ENIAC and is built with discrete transistors.*

This miniaturization trend continued, and in 1971 the first microprocessor (the 4004) was introduced. Microprocessors contain the major computation and control sections of a computer, called the *Central Processing Unit* (CPU), on a single integrated circuit "chip." Microprocessors are often called *Microprocessing Units* (MPUs). A microprocessor chip contains thousands of gates and is called a *Large Scale Integration* (LSI) device. LSI memory devices were also developed that store thousands of bits of digital information on a single IC. These two LSI devices made it possible to drastically reduce the size and cost of small computers. Microprocessors have made it practical to build dedicated computers into many small, inexpensive products.

Hand-Held Programmable Calculator. *Integrated circuit technology made possible a hand-held calculator more powerful than the desktop calculator pictured above.*



Single-Chip Microcomputer. *Less than a quarter of an inch on a side, this tiny chip contains virtually all the electronics for a small computer system. (Photo Courtesy Intel Corp.)*

WHY MICROPROCESSORS ARE USED

Microprocessors are now being used in many products which were previously built with random logic. Microprocessor-based designs are usually less expensive and have many fewer components than the designs that they replace. Small microprocessor systems may be built with one or two ICs, at a cost of under ten dollars. These can often replace boards with dozens of simpler ICs. Because the number of discrete components and interconnections is greatly reduced, reliability is also improved.

This reduction in size is also possible using custom integrated circuits instead of microprocessors. However, the design of a custom IC can be an extremely complex and expensive process, often costing well over $100,000. This expense can be justified only for high volume products where the development cost can be spread over many thousands of units. The microprocessor allows standard ICs to be used to achieve the same miniaturization. The customizing takes the form of the program stored in the memory. Producing a standard memory with a custom program stored in it is a relatively inexpensive process.

The flexibility and power of microprocessor-based systems makes many sophisticated features possible, which in the past were impractical. For example, microprocessor-based systems can often test themselves to a considerable extent and provide appropriate error messages. Instruments such as digital voltmeters can provide functions such as automatic averaging of a number of measurements, addition or subtraction of an offset value from each measurement, and self-calibration. The microprocessor also makes practical the use of a keyboard instead of multiposition switches on the front panel. Another feature is the capability for complete remote control.

Microprocessor-Based Digital Voltmeter. *The microprocessor provides functions such as self-calibration, averaging of a number of measurements, and automatic addition of an offset value.*

One of the most visible products to emerge from microprocessor technology is the electronic cash register or *Point of Sale* (POS) terminal. By replacing older electromechanical machines, reliability has been improved and many new features have become possible. By assigning each item to be sold an identification number, which the operator either types on the keyboard or reads from a bar code, inventory can be kept automatically. Sales tax can also be automatically added. It is even possible for the customer's bank account number to be entered, and the money automatically transferred from his bank account to the store's account.

Another example of the convenience which can be provided by microprocessor-based products is the computing scale. The operator enters the price per pound, and the scale weighs the item and displays the weight and the cost. The scale can also subtract the weight of the container.

Microprocessor-Based Scale. *The user enters the price per pound, and the scale computes the total price. The weight of the container can also be automatically subtracted. (Photo Courtesy Toledo Scale)*

All of these products may have been possible without microprocessors, but they would be so complex and expensive that they would be impractical. Microprocessor-based systems have these capabilities because so much complexity has been placed inside each IC. Product designers do not have to worry about the detailed construction of the ICs, and the size and reliability problems associated with complex systems are avoided. Furthermore, by placing the control in software, design changes are easy to make. The microprocessor has indeed revolutionized the design of many products.

A BASIC MICROPROCESSOR SYSTEM

Consider a system with a keyboard and a numeric display, as in a pocket calculator. When a key is pressed, the corresponding number should appear on the display. This system is a natural application for a microprocessor, and is in many ways similar to the Microprocessor Lab.

Figure 1-1 shows the block diagram of a system for doing this. The microprocessor (also called the *processor*) is the "brains" of the system. It contains all of the logic to recognize and execute the list of instructions *(program)*. The *memory* stores the program, and may also store data. The fold-out inside the back cover shows these components on the Microprocessor Lab (μLab) board.

The microprocessor needs to exchange information with the keyboard and display. The *input port*, from which the processor can read data, connects the processor to the keyboard. The *output port*, to which the processor can send data, connects the processor to the display.

The blocks within the microcomputer are interconnected by three buses. A *bus* is a group of wires which connect the devices in the system in parallel. The microprocessor uses the *address bus* to select memory locations or input and output ports. You can think of the addresses as post office box numbers; they identify which locations to put information into or take information out of.

Once the microprocessor selects a particular location via the address bus, it transfers the data via the *data bus*. Information can travel from the processor to the memory or an output port, or from an input port or memory to the processor. Note that the microprocessor is involved in all data transfers. Data usually does not go directly from one port to another, or from the memory to a port.

The third bus is called the *control bus*. It is a group of signals which are used by the microprocessor to notify memory and I/O devices that it is ready to perform a data transfer. Some signals in the control bus allow I/O or memory devices to make special requests from the processor. The control bus is not apparent on the μLab board because it connects directly to the control logic, which generates control signals for each device in the system.

A single digit of binary information (1 or 0) is called a *bit* (a contraction of *binary digit*). One digital signal (high or low) carries one bit of information. Microprocessors handle data not as individual bits, but as groups of bits called *words*. The most common microprocessors today use eight-bit words, which are called *bytes*. These microprocessors are called eight-bit processors. For an eight-bit processor, *byte* and *word* are often used interchangeably. Be aware, however, that *word* is also used to mean a group of sixteen or more bits.

The address and data buses may also be seen on the μLab board. The data bus is a group of eight lines, and the address bus has sixteen lines.

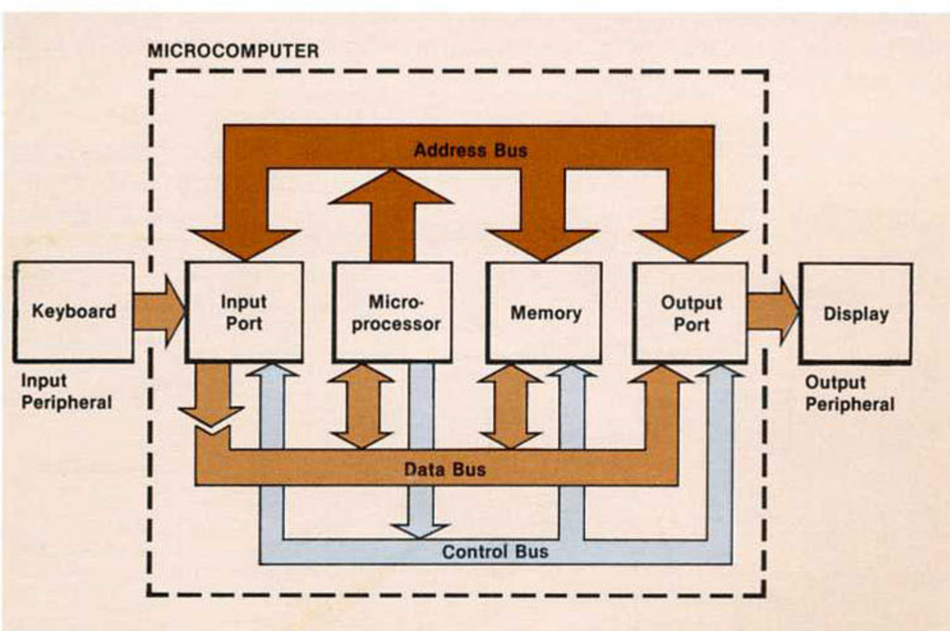


Figure 1-1. Basic Microprocessor System

PROGRAMS

To direct the system to perform the desired task, an appropriate list of instructions is required. For example:

1. Read data from the keyboard.
2. Write data to the display.
3. Repeat (go to step 1).

For the microprocessor to perform a task from a list of instructions, the instructions must be translated into a code that the microprocessor can understand. These codes are then stored in the system's memory. The microprocessor begins by reading the first coded instruction from the memory. The microprocessor decodes the meaning of the instruction and performs the indicated operation. The processor then reads the instruction from the next location in memory and performs the corresponding operation. This process is repeated, one memory location after another.

Certain instructions cause the microprocessor to *jump* out of sequence to another memory location for the next instruction. The program can therefore direct the microprocessor to return to a previous instruction in the program, creating a *loop* which is repeatedly executed. This enables operations which must be repeated many times to be performed by a relatively short program.

PERIPHERALS

A complete microprocessor system, including the microprocessor, memory, and input and output ports is called a *microcomputer*. The devices connected to the input and output ports (the keyboard and display for example) are called *peripherals*, or *Input/Output* (I/O) devices. The peripherals are the system's interface with the user. They may also connect the microcomputer to other equipment. Storage devices such as tape or disc drives are also referred to as peripherals.

An example of a microprocessor application from the instrumentation field is the microprocessor-based digital voltmeter (see Figure 1-2). Its input peripherals are an analog-to-digital converter and the range and function selector switches. The output peripheral is a digital display. The basic microcomputer is the same, whether the application is a calculator or a voltmeter; the difference is in the peripherals and the program.

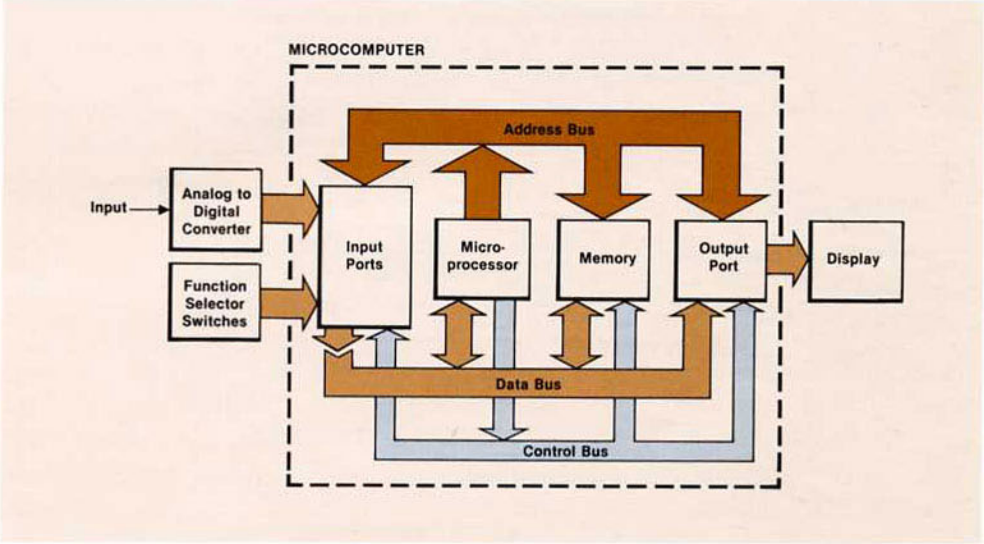
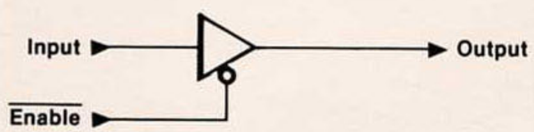


Figure 1-2. Microprocessor-Based Digital Voltmeter

THREE-STATE DRIVERS

All devices in the microprocessor system exchange information with the microprocessor over the *same* set of wires (the data bus). The microprocessor selects one device to place data on the data bus and disconnects the others. It is the *three-state* output capability of the devices on the bus that enables the processor to selectively turn devices on and off.



$\overline{\text{Enable}}$	Input	Output
0	0	0
0	1	1
1	0	floating
1	1	floating

0 = low 1 = high

Figure 1-3. Three-State Driver

Figure 1-3 shows the symbol and truth table for a three-state buffer (often called a *three-state driver*). The buffer has an output enable in addition to the usual input and output. When the enable is low, the buffer acts just as an ordinary buffer. The signal at the input is transferred to the output. When the enable is high, on the other hand, the output of the device is essentially disconnected.

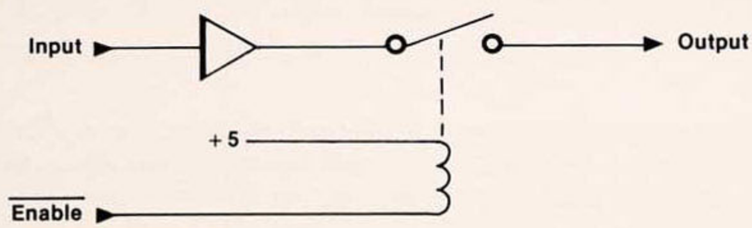


Figure 1-4. Conceptual Equivalent of Three-State Driver

Figure 1-4 shows a conceptual equivalent circuit which generates the open state using a relay. The disabled (open) output state is often called the *high impedance* state. Figure 1-5 shows a schematic of a typical three-state output.

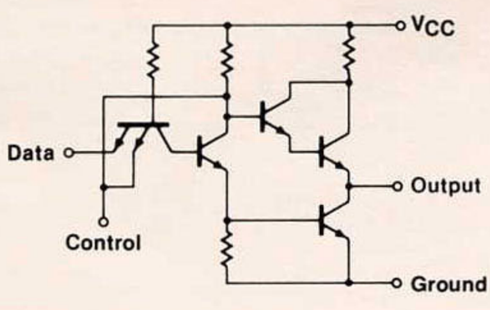


Figure 1-5. Schematic of Typical Three-State Output

Three-state drivers are important because they allow many devices to share a single data line. The circuit shown in Figure 1-6 allows any one of three different signals to drive one output. Only one driver's enable line may be low, and that device drives the output. If more than one driver were enabled, they would both try to drive the output. This condition is not allowed because the logic state of the output would be unpredictable.

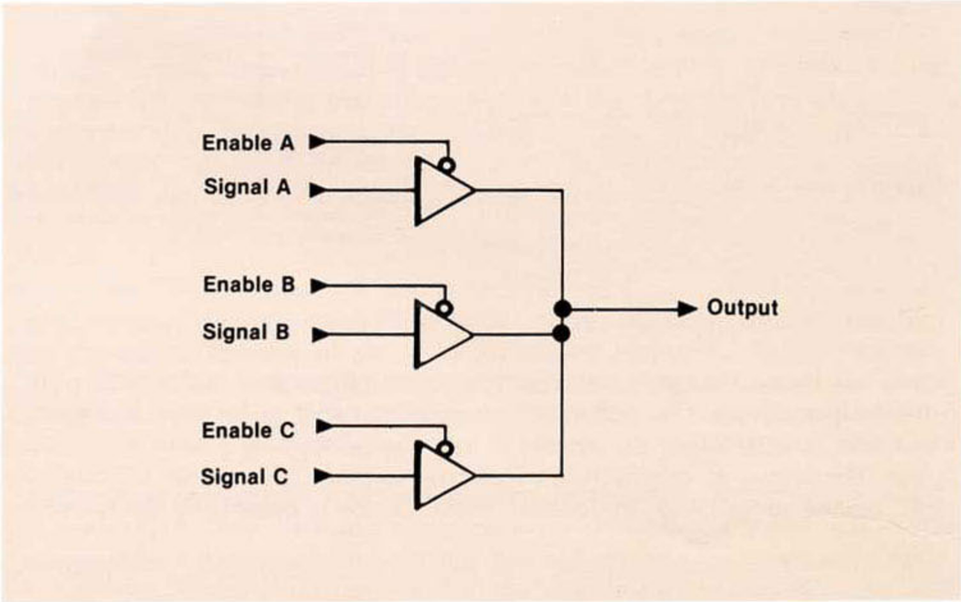


Figure 1-6. Circuit Showing Several Signals Sharing Single Data Line

Many devices, including microprocessors and memories, contain internal three-state drivers. These ICs have an output enable, often called *Chip Select* (CS) or *Chip Enable* (CE), which controls their output drivers.

Figure 1-7 shows how three-state drivers are used in microprocessor systems. All devices which put data on the data bus have three-state drivers on their outputs. The microprocessor generates control signals (part of the control bus) to enable the three-state drivers of the device from which it wants to read data. The three-state drivers of the other devices are disabled.

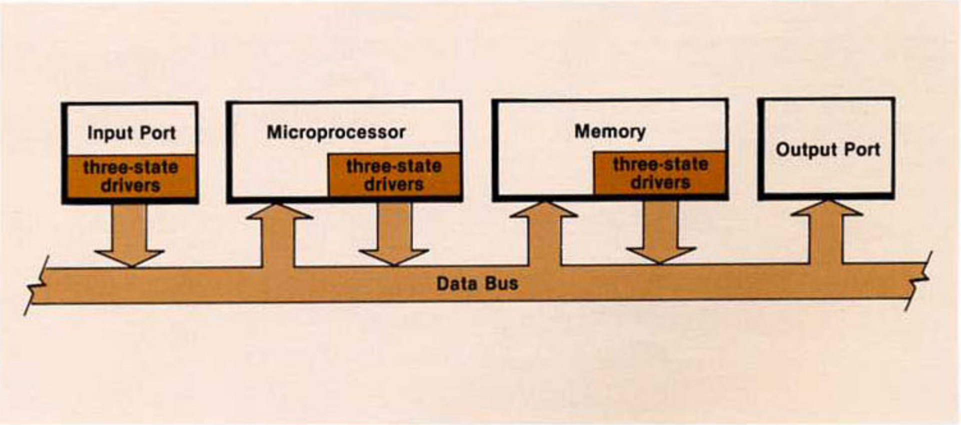
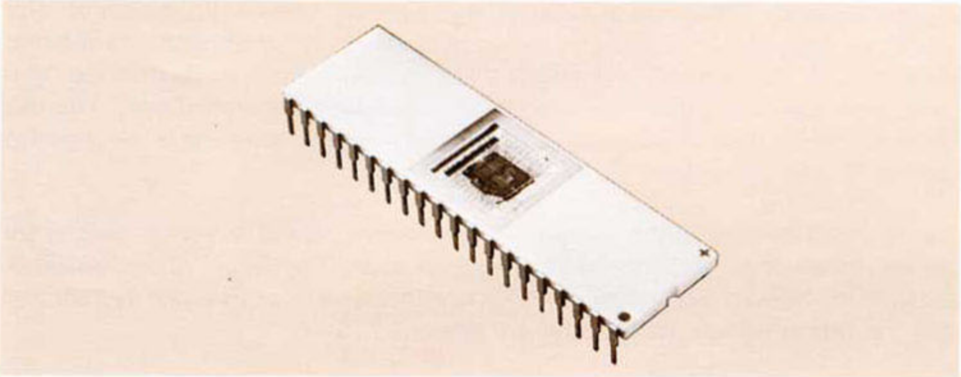


Figure 1-7. Three-State Drivers in Microprocessor System



8085 Microprocessor in Package with Lid Removed. *Forty thin wires connect pads on the chip to connection points on the package. (Photo Courtesy Intel Corp.)*

THE MICROPROCESSOR

Figure 1-8 shows the basic signals that connect to a typical microprocessor. There are sixteen address outputs which drive the address bus, and eight data pins which connect to the data bus. The data pins are *bidirectional*, which means that data may go into or out of them. $\overline{\text{READ}}$ and $\overline{\text{WRITE}}$ are the control signals that coordinate the movement of data on the data bus.

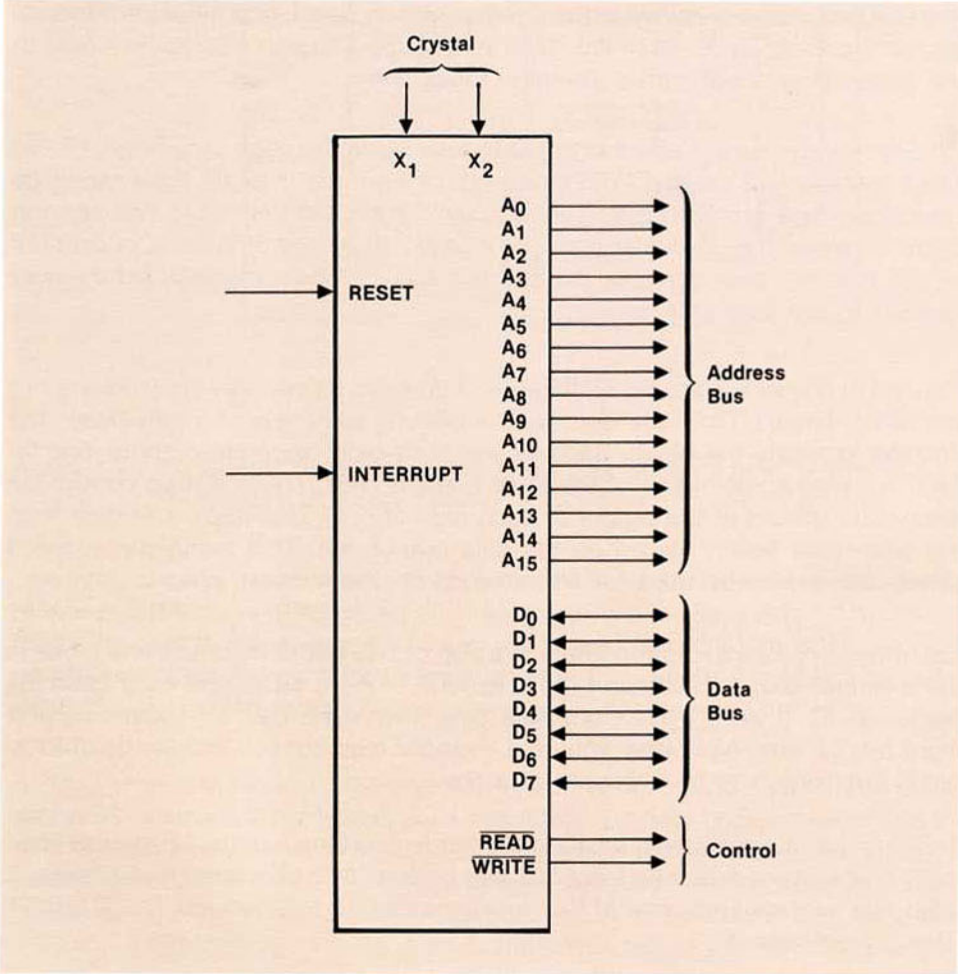


Figure 1-8. Basic Microprocessor Signals

The two signals shown on the left of the diagram provide additional control functions. The RESET input is used to initialize the microprocessor's internal circuitry. The INTERRUPT input allows the microprocessor to be diverted from its current task to another task which must be performed immediately. The use of these signals, plus others which have not been mentioned here, is described in section III, Microprocessor System Hardware.

The two connections at the top are for an external crystal, which is used to set the frequency of an oscillator in the microprocessor. The output of this oscillator is called the system *clock*. The clock synchronizes all devices in the system and sets the rate at which instructions are executed.

MEMORIES

Microprocessor systems usually use integrated circuit memories to store programs and data. They can store many bits of data in a single IC. Currently, devices are available with capacities of over 65,000 bits on one chip. A 65,536-bit memory can store over eight thousand alphanumeric characters, or about three pages of this text on a piece of silicon about a third of an inch square.

The simplest memory device is the *flip-flop*, which stores one bit of information. *Registers* contain up to eight flip-flops on a single IC, each with its own data in and data out pins but with a common clock line.

LSI technology made it possible to put thousands of flip-flops on a single IC, but a new problem was created. With thousands of flip-flops on an IC, there cannot be a separate data pin for each. The solution to this problem is to use address inputs to select the particular memory location (flip-flop) of interest. A *decoder* on the memory chip decodes the address and connects the selected memory location to the data pins.

Figure 1-9 shows a conceptual diagram of an eight-bit memory (most memories are much larger). Only the data output circuits are shown for simplicity. The decoder converts the binary address inputs to eight separate outputs, one for each possible combination of the three address lines. These signals control the three-state drivers at the output of each memory cell (flip-flop). The data from the addressed cell is placed on the data output line. This technique allows a single data pin to be used for all locations on the memory chip.

Each memory location can contain a group of bits rather than just one bit as in the example above. Each can hold one, four, or eight bits, depending upon the particular IC. If the IC has eight data pins, then each memory location stores eight bits of data. Note that while the memory may contain thousands of locations, only one may be accessed at a time.

The number of addressable locations depends upon the number of address lines. With one address line, two locations can be selected: address 0 and address 1. With two address lines, one of four locations can be selected: 00, 01, 10 and 11. The general rule is:

Number of locations $= 2^N$
Where N = number of address lines

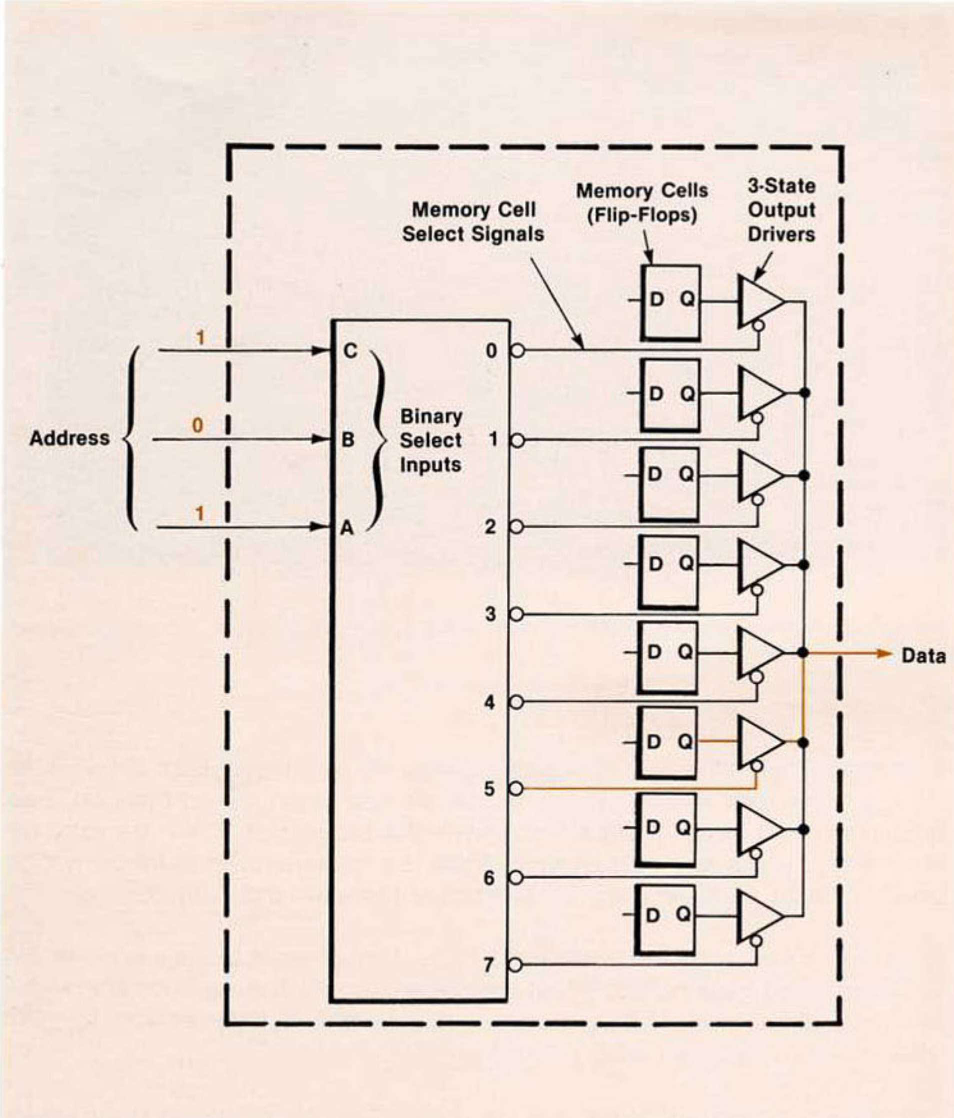
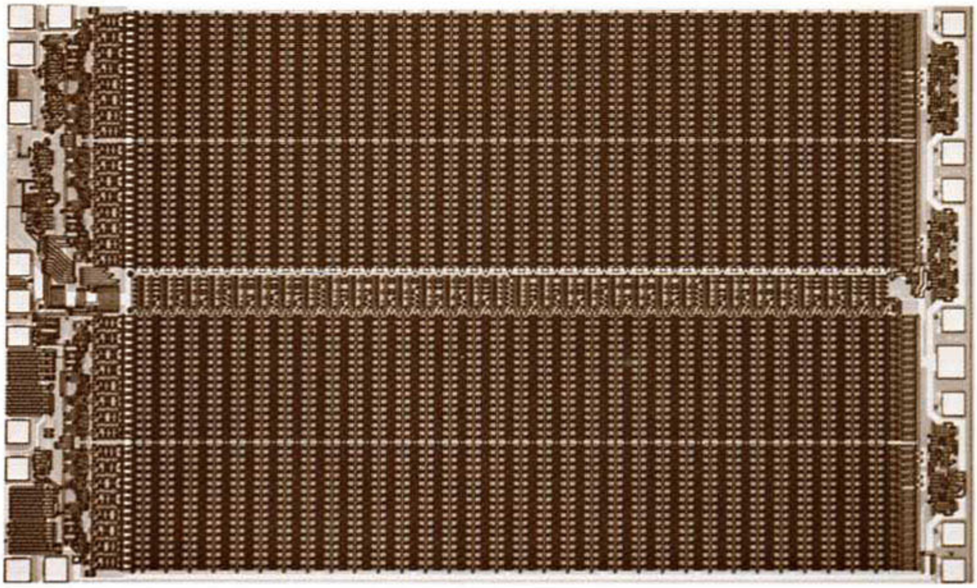


Figure 1-9. Conceptual Diagram of Eight-Bit Memory, Showing How All Memory Cells Share Single Data Line. Location 5 (101 binary) is shown selected.

ROMS AND RAMS

The memory ICs used with microprocessors fall into two broad categories: *ROMs* and *RAMs*. A ROM (Read Only Memory) is a memory which can only be read. The data is programmed into it at the time of manufacture, or by a special programming procedure prior to installation in the circuit. A program recorded into a ROM is often referred to as *firmware*.

A RAM (Random Access Memory) is a memory into which data can be stored and then retrieved. RAM is actually a misnomer; random access means that the time to access any memory location is the same, a characteristic also present in ROMs. Read/Write (R/W) memory is a more accurate term for what are usually called RAMs, but RAM is widely used to mean integrated circuit read/write memory. A digital tape recorder is an example of a memory which is not random access, since the time to access a particular location depends upon the position of the tape.



Photomicrograph of 4,096-Bit RAM. *The chip is 0.225 × 0.142 inches. (Photo Courtesy Intel Corp.)*

An important characteristic of semiconductor RAMs is that they are *volatile*: they lose their data when power is turned off, and when turned back on, they contain unknown data. ROMs do not have this problem, so they are used for permanent program and data storage. Since the contents of a ROM cannot be modified, RAMs must be used for temporary program and data storage.

Figure 1-10 shows a ROM containing 2,048 words of eight bits each, or 16,384 bits. When using large numbers that are powers of two, K is often used to mean 1,024 (2^{10}). Thus, this memory has 2K bytes or 16K bits. Since each location contains eight bits, it is called a 2K × 8 ROM.

When the Chip Select ($\overline{CS}$) input is low, the ROM's output drivers are enabled. When $\overline{CS}$ is high, the data outputs are in the high impedance state. The three-state outputs allow the data lines of many memory devices to be connected together, with one device selected by bringing its $\overline{CS}$ input low.

Figure 1-11 shows a 1K × 8 RAM. This RAM contains 1,024 locations of eight bits each. The data lines are bidirectional, since data can go into or out of the memory. RAMs have an additional control line called $\overline{WRITE}$. To store data in the RAM, an address is selected, the data is placed on the data lines, and the $\overline{WRITE}$ line is brought low. When the data and address are all set, the chip select is pulsed, and the data is stored in the memory.

The write line determines the direction of the data flow. The write line is usually active low, and is often called RD/$\overline{WR}$ (or R/$\overline{W}$). This notation indicates that if the signal is high, a read is performed, and if it is low, a write is performed. Note that this input has no effect unless the chip select is true.

ROMs and RAMs come in many different sizes (with different numbers of words and different numbers of bits per word) and many types. Lesson 9 contains additional information on ROMs and RAMs.

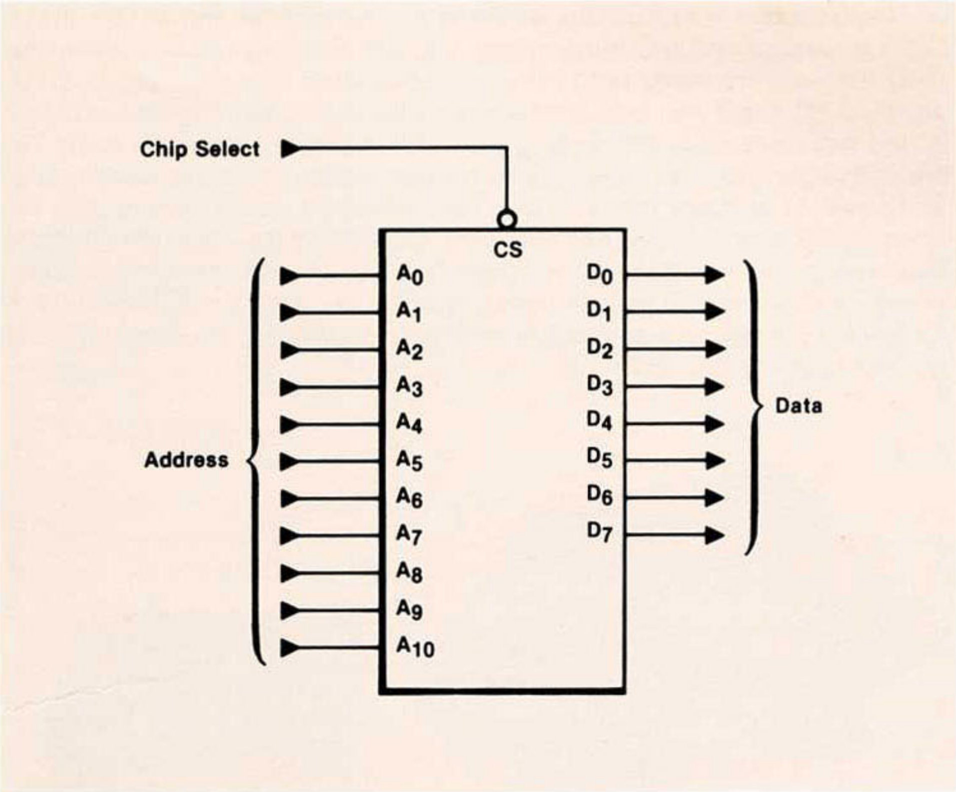


Figure 1-10. 2K × 8 ROM

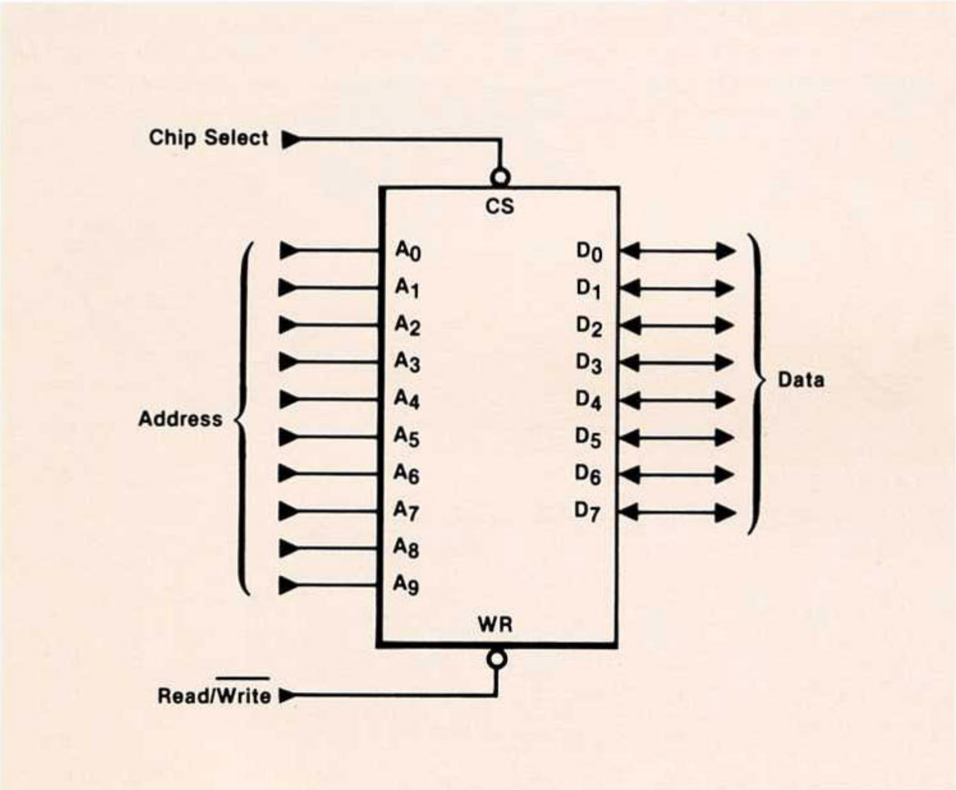
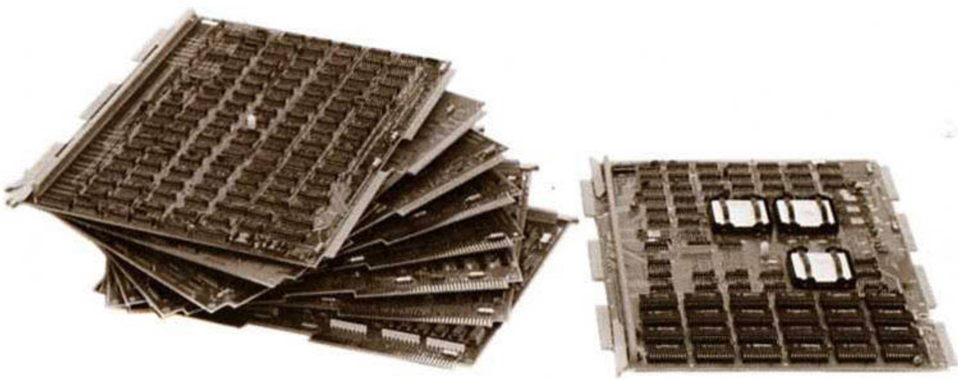


Figure 1-11. 1K × 8 RAM

MICROCOMPUTERS AND MINICOMPUTERS

A microcomputer is functionally similar to a minicomputer, and, in fact, the distinction between the two is becoming less and less clear. A microcomputer's CPU (Central Processing Unit) is the microprocessor. A minicomputer's CPU is usually a PC board with dozens of less complex, but faster integrated circuits on it. The main functional difference is that minicomputers are usually faster. They are also larger and more expensive. As microprocessors have increased in speed and power to compete with the older minicomputers, new minicomputers have been developed which are even faster and more powerful. These new minicomputers often use microprocessors internally. Thus, while the basic distinctions of speed, power, and size remain, the exact boundary is becoming vague. Microcomputers are now finding applications in systems where a minicomputer would be far too bulky and expensive.



One Board Replaces Eight. *On the left is the set of eight circuit boards which comprise the CPU for the HP 3000 Series II computer. The boards are built with standard small and medium scale TTL ICs. On the right is a complete CPU which performs exactly the same function with a single board. The three large squares are custom LSI ICs which make the size reduction possible. The single-board CPU is used in the Series 33 computer.*